MSA: Jointly Detecting Drug Name and Adverse Drug Reaction Mentioning Tweets with Multi-Head Self-Attention

Chuhan Wu
Electronic Engineering
Tsinghua University
wuch15@mails.tsinghua.edu.cn

Fangzhao Wu
Microsoft Research Asia
Beijing, China
wufangzhao@gmail.com

Zhigang Yuan
Electronic Engineering
Tsinghua University
ljx16@mails.tsinghua.edu.cn

Junxin Liu
Electronic Engineering
Tsinghua University
ljx16@mails.tsinghua.edu.cn

Yongfeng Huang
Electronic Engineering
Tsinghua University
yfhuang@tsinghua.edu.cn

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

ABSTRACT

Twitter is a popular social media platform for information sharing and dissemination. Many Twitter users post tweets to share their experiences about drugs and adverse drug reactions. Automatic detection of tweets mentioning drug names and adverse drug reactions at a large scale has important applications such as pharmacovigilance. However, detecting drug name and adverse drug reaction mentioning tweets is very challenging, because tweets are usually very noisy and informal, and there are massive misspellings and user-created abbreviations for these mentions. In addition, these mentions are usually context dependent. In this paper, we propose a neural approach with hierarchical tweet representation and multi-head self-attention mechanism to jointly detect tweets mentioning drug names and adverse drug reactions. In order to alleviate the influence of massive misspellings and user-created abbreviations in tweets, we propose to use a hierarchical tweet representation model to first learn word representations from characters and then learn tweet representations from words. In addition, we propose to use multi-head self-attention mechanism to capture the interactions between words to fully model the contexts of tweets. Besides, we use additive attention mechanism to select the informative words to learn more informative tweet representations. Experimental results validate the effectiveness of our approach.

KEYWORDS

Twitter, Drug Name, Adverse Drug Reaction, Self-Attention

ACM Reference Format:
Chuhan Wu, Fangzhao Wu, Zhigang Yuan, Junxin Liu, Yongfeng Huang, and Xing Xie. 2019. MSA: Jointly Detecting Drug Name and Adverse Drug Reaction Mentioning Tweets with Multi-Head Self-Attention. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM ’19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3290980

WSDM ’19, February 11–15, 2019, Melbourne, VIC, Australia
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290980

Table 1: Several examples of detecting tweets mentioning drug names (DN) and adverse drug reactions (ADR).

Task	Tweet	Mention?
DN	I take Vitamin C after meal	yes
	Vitamin C is good for health	no
	Oranges are good for health	no
ADR	addreal makes me sleepless	yes
	No aspirin pills and I’m quite pain	no
	I feel headache after drinking coffee	no

1 INTRODUCTION

Social media platforms such as Twitter have attracted a huge number of users for information sharing and dissemination [23, 29]. Many users share information about health, drugs and treatments in their tweets. Automatically detecting tweets mentioning drug names (DNs) and adverse drug reactions (ADRs) at a large-scale is an important task in the natural language processing and data mining fields, and has many important applications such as pharmacovigilance [25]. For example, mining possible adverse reactions from tweets can help drug manufacturers to guarantee the safety of drugs and reduce their detrimental impact on patients [3]. Compared with ADR detection from electronic health records [31] and clinical reports [1], tweets can easily be collected in real-time and the number of tweets mentioning drug names and adverse drug reactions is huge. Thus, detecting the tweets mentioning drug names and adverse drug reactions has the potential to help discover serious or unknown consequences of drug uses that are not covered by medical records [14].

The task of detecting drug names and adverse drug reactions mentioning tweets has become a hot research topic and has been extensively studied in recent years [33]. It is usually formulated as a binary text classification task. Several examples are shown in Table 1. Machine learning techniques are widely used in this task. For example, Sarker et al. [25] applied support vector machine (SVM) to detect adverse drug reactions mentioning tweets. They use various features such as n-gram features, ADR lexicon matches, sentiment features and topic modeling keywords. Zhang et al. [34] applied maximum entropy (ME) algorithm to this task and they used an ensemble of models trained with different features, such as lexicon features extracted from ADR lexicons, n-gram features and word embeddings. However, these methods usually rely on heavy manual feature engineering, which requires a large amount of expert knowledge and time to craft. In recent years, with the development of deep learning, many neural network based methods such as convolutional neural network (CNN) [17] are introduced to this task. For example, Huynh et al. [14] proposed to use CNN with max pooling techniques to classify whether tweets contain adverse drug reaction mentions or not. Lee et al. [21] applied the CNN framework proposed by Johnson et al. [15] to detect tweets mentioning adverse drug reactions. In their approach, word embeddings are enhanced by pretraining on unlabeled data. These deep learning based methods can reduce the efforts of manual feature engineering and can achieve better performance than traditional methods such as SVM. However, since there are massive misspellings and user-created abbreviations which are often out-of-vocabulary, it is very difficult for these methods to learn high-quality representations of tweets from the words. In addition, these methods cannot model the interactions between the words within tweets, which are important to capture the contexts of tweets.

Our approach is motivated by following observations. First, tweets are usually very noisy and informal, and are full of misspellings and user-created abbreviations. For example, in the tweet "addreal makes me sleepless" in Table 1, the word "addreall" is misspelled as "addreal". Second, many drug name and adverse drug reaction mentions are context-dependent. For instance, "Vitamin C" is a drug name in the tweet "I take Vitamin C after meal", but is a non-drug mention in the tweet "Vitamin C is good for health". Third, different words in the same tweet usually have different informativeness for drug name and adverse drug reaction mentioning tweet detection. For example, in the tweet "running out of aspirin makes me in huge pain today", the words "aspirin" and "pain" are informative to the detection of adverse drug reaction mentioning tweets, but the word "today" is not very informative. Fourth, it is a common observation that a tweet which mentions adverse drug reactions may also mention drug names, and a tweet mentioning drug names may also provide some information on adverse drug reactions. Therefore, the task of detecting drug name mentioning tweets and the task of detecting adverse drug reaction mentioning tweets have inherent relatedness with each other, and jointly training models for these two tasks has the potential to improve the performance of both tasks.

Motivated by the aforementioned observations, we propose a neural approach based on hierarchical tweet representation and multi-head self-attention (MSA) to jointly detect tweets mentioning drug names and adverse drug reactions. In order to alleviate the influence of massive misspellings and user created abbreviations in tweets, we propose to use a hierarchical tweet representation model to first learn representations of words from their characters and then learn representations of tweets from words. In addition, since drug name and adverse drug reaction mentions are often context-dependent, we propose to use multi-head self-attention mechanism to fully model the contexts of tweets. Besides, since different words have different informativeness for detecting tweets mentioning drug names and adverse drug reactions, we use additive attention mechanism to select the informative words in tweets to build more informative representations of tweets. Experimental results on two benchmark datasets show that our approach can effectively improve the performance of drug name and adverse drug reaction mentioning tweets detection, and consistently outperform many baseline methods.

2 RELATED WORK

ADR mention detection is a hot research topic in the natural language processing and data mining fields. Traditional ADR mention detection methods focus on Electronic Health Records (EHR) [6, 31] and clinical reports [1, 10]. However, since EHRs and clinical reports on specific ADRs are not abundant and are difficult to collect, it is difficult to build a robust ADR detection model based on EHRs and clinical reports [12]. In addition, the ADRs covered by these resources are also very limited.

With the development of social platforms, detecting ADRs and DNs at a large scale on social media platforms such as Twitter has received increasing interest from researchers [25]. It is usually formulated as a binary text classification task, and has been extensively studied in recent years [26, 28, 33]. Existing methods for this task are mainly based on machine learning techniques [24, 27, 34]. For example, Sarker et al. [25] proposed to use SVM to classify whether a tweet contains an ADR mention based on various features, such as n-gram features, sentiment features based on SentiwordNet [5], lexicon features extracted from ADR lexicons [19] and topic features extracted by Latent Dirichlet Allocation (LDA) [2]. Jonnagaddala et al. [16] applied SVM to this task based on features derived from Sarker et al. They also use linguistic features such as part-of-speech (POS) tags of words. Zhang et al. [34] applied maximum entropy (ME) algorithm to this task. They ensemble different models, including ADR lexicon matching classifier, n-gram based ME classifier and word embedding based ME classifier. Kiritchenko et al. [18] also employed SVM in this task. They proposed to incorporate additional features such as word embeddings, word cluster, negation and the number of hashtags and emoticons. However, the major disadvantage of the aforementioned methods is the heavy dependency on manual feature engineering, which usually needs a large amount of domain knowledge to craft. Besides, due to the dependency of bag-of-word features, these methods usually cannot effectively capture the contexts and orders of words in texts, which are both very useful to DN and ADR detection.

In recent years, many deep learning based methods have been proposed for the DN and ADR detection task [14, 21, 22]. For example, Magge et al. [22] applied CNN to this task. They used a cost sensitive loss function and the random under-sampling technique to solve the problem of imbalanced class distribution. Lee et al. [21] introduced the semi-supervised CNN model proposed in [15] to this task. They use many different types of corpus to enhance the word embeddings in their models. However, these methods simply aggregate all words within texts together to build their representations, and cannot effectively distinguish informative contexts from the uninformative ones. There have been several studies to incorporate attention mechanism in this task to address this problem. For example, Huynh et al. [14] applied attention mechanism after CNN to compute an attention weight for each word to represent the importance of it. Han et al. [11] proposed to compute multiple attention weights for the word representations obtained by CNN. However, the performance of these methods is unsatisfactory. The improvements brought by the attention networks in these approaches are either marginal or even negative. In addition, there are many non-standard languages and new words in tweets, and these words are often out-of-vocabulary (OOV). However, the aforementioned methods can only utilize the information of the words in tweets, and their performance may be affected by the massive typos and user-created abbreviations in tweets. Different from these existing methods, in our approach we use a hierarchical tweet representation model to learn representations of words from their characters and tweet representations from the representations of their words. Our approach can exploit character information from the noisy tweets to enhance word representations by capturing the patterns of character combinations. In addition, we propose to apply multi-head self-attention mechanism to build high-quality representations of words by modeling the interactions of a word between all words within a tweet. Besides, since the tasks of detecting tweets mentioning DN and mentioning ADR may have inherent relatedness, we propose to train both tasks jointly to exploit the mutual information between the two tasks. Experimental results on the benchmark datasets show that our approach outperforms existing baseline methods and can achieve satisfactory performance on the detection of DN and ADR mentioning tweets.

3 OUR APPROACH

In this section, we will introduce our neural approach using multi-head self-attention (MSA) to jointly detect DN and ADR mentioning tweets. There are three modules in our MSA model. The first one is a word representation module, which aims to build the contextual representations of words from the original characters within them. The second one is a tweet representation module, which aims to build the representations of a tweet from the representations of its words. The third one is a tweet classification module, which classifies whether a tweet contains DN or ADR mentions based on its representations. In addition, we apply multi-head self attention mechanism to build high-quality contextual representations of words by capturing interactions between words. Besides, we incorporate additive attention mechanism to highlight the important contexts to build the representations of tweets. The architecture of our MSA model is illustrated in Fig. 1. We will introduce the details of each module of our approach in the following sections.

3.1 Word Representation

Since words are usually basic units to condense meanings, building word representations is usually a prerequisite for neural network based methods. However, there are massive misspelling (e.g., "vtamin" for "vitamin") and user-created abbreviations (e.g., "mop" for "morphine") of drug names and adverse drug reactions in tweets. Therefore, many DN and ADR mentions are out-of-vocabulary and it is difficult to build high-quality representations of these words directly. Motivated by these observations, we propose to learn word representations from their characters first. There are three submodules in the word representation module. We will introduce each of them in details.

The first one is a character embedding layer. It is used to transform the sequence of characters within each word into a sequence of low-dimension dense vectors. We denote the character sequence of i_{th} word as $w_i = [C_{i,1}, C_{i,2}, ..., C_{i,N}]$, where N is the length of this word. It will be converted into a vector sequence $\mathbf{E}_i^c = [\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, ..., \mathbf{e}_{i,N}]$ via a character look-up matrix $\mathbf{M^c} \in \mathcal{R}^{V \times D}$, where V denotes the character vocabulary size and D denotes the dimension of character embedding.

The second one is a character-level convolutional neural network (CNN). CNN is effective to capture local context information [17]. Usually, the local character combinations can be important clues for DN and ADR detection because of the rules in chemical nomenclature. For example, many drug names contain the string "benz", which often indicates that there are benzene rings in the molecular structures of drugs. Thus, capturing local contextual information of characters is useful to detect tweets mentioning DNs and ADRs. The CNN layer is used to construct the contextual representations of characters by capturing local information as follows:

$$h_{i,j} = \text{ReLU}(\mathbf{U}_c \times \mathbf{e}_{i,(j-w):(j+w)} + b_c), \tag{1}$$

where $\mathbf{e}_{i,(t-w):(t+w)}$ represents the concatenation of the character embedding vectors between position $t - w$ and $t + w$, $\mathbf{U}_c$ and b_c are the kernel and bias parameters of CNN, $2w + 1$ is the window size of CNN filters and ReLU is the non-linear activation function [7]. We apply the CNN layer to all character sequences within a tweet, and the final contextual representations of the character $C_{i,j}$ is the concatenation of the outputs of multiple filters, which is denoted as $\mathbf{h}_{i,j}$. In order to build the character-based representations of words, we apply the max pooling operation to the feature maps obtained by CNN to keep the most salient information from the character representations. We denote the character-based representations of the i_{th} word as $\mathbf{p}_i$.

The third one is word embedding. It is used to enhance the word representations by incorporating rich semantic information extracted from a large collection of tweets. We denote the word sequence as $s = [w_1, w_2, ..., w_M]$, where M is the length of the tweet. It will be converted into a vector sequence $\mathbf{E}^w = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M]$ using a word embedding matrix $\mathbf{M^w} \in \mathcal{R}^{V' \times D'}$, where V' and D' respectively denote the vocabulary size and the word embedding dimension. The final representations $\mathbf{c}_i$ of each word is the concatenation of the character-based representations and word embeddings, i.e., $\mathbf{c}_i = [\mathbf{p}_i; \mathbf{e}_i]$. We denote the output sequence of word representation vectors as $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_M]$.

3.2 Tweet Representation

The tweet representation module aims to learn the hidden representations of tweets from the representations of words within them. It has three sub-modules, and the details are described as follows:

The first one is a bi-directional long short term memory (Bi-LSTM) network [9]. Long-distance information is important for the detection of tweets mentioning DNs and ADRs. For example, the tweet "adderall and coffee make me feel like i was on the verge of a heart attack" mentions a possible ADR "heart attack", which has a long distance to the drug name "adderall". LSTM is an effective neural model to capture such long-distance information [20]. Since both past and future information may be useful to build the contextual representations of words, we use Bi-LSTM network in our approach. It scans the sequence of word representation vectors

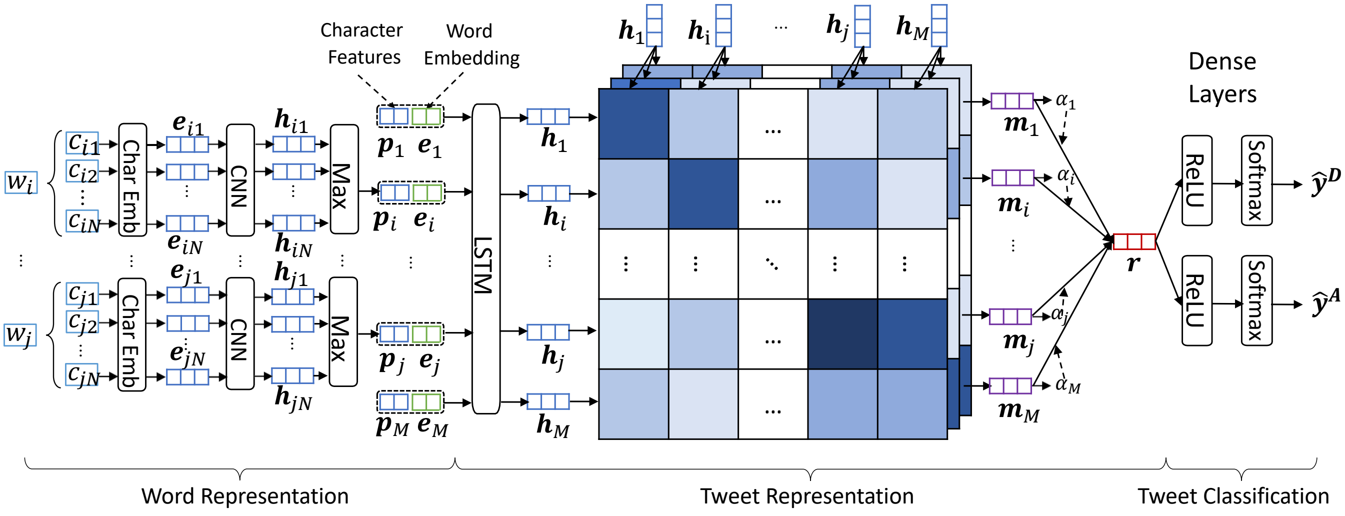


Figure 1: The framework of our *MSA* model.

in both directions, and outputs the hidden states at each position. We denote the output hidden states of all words within a tweet as $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_M]$, which is obtained by $\mathbf{H} = LSTM([\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_M])$.

The second one is a multi-head self-attention network. Usually, many DN and ADR mentions are context-dependent, and the interactions between words are important to detect the tweets mentioning DNs and ADRs. For example, in the tweet "penicillin makes me allergy", the interaction between "penicillin" and "allergy" is an important clue for ADR detection. Self-attention is an effective method to capture the informative interactions between words in texts [30]. In addition, a word may interact with multiple words. For example, in the tweet "I drink much coffee with adderall and I'm extremely excited", the interaction of "adderall" with "coffee" and the interaction of "adderall" with "excited" are both important to detect the ADR mentions. Therefore, we propose to use multi-head self-attention mechanism [30] to build high-quality contextual representations of words by drawing their interactions with multiple words jointly. In the multi-head self-attention network, the representation vector $\mathbf{m}_{i,j}$ of the j_{th} word produced by the i_{th} attention head is computed by a weighted summation of all word representation vectors in $\mathbf{H}$, which is formulated as follows:

$$\hat{\alpha}^i_{j,k} = \mathbf{h}_j^T \mathbf{U}_i \mathbf{h}_k, \tag{2}$$

$$\alpha^i_{j,k} = \frac{\exp(\hat{\alpha}^i_{j,k})}{\sum_{m=1}^{M} \exp(\hat{\alpha}^i_{j,m})}, \tag{3}$$

$$\mathbf{m}_{i,j} = \mathbf{W}_i(\sum_{m=1}^{M} \alpha^i_{j,m} \mathbf{h}_m), \tag{4}$$

where $\mathbf{U}_i$ and $\mathbf{W}_i$ are the projection parameters of the i_{th} self-attention head, $\alpha^i_{j,k}$ represents the relative importance of the interaction between the j_{th} and k_{th} words. In this way, the hidden representation of each word is constructed from the hidden representations of all words in a tweet by modeling the interactions between this word with all words. The multi-head representation $\mathbf{m}_j$ of the j_{th} word is the concatenation of the representation vectors produced by h separate self-attention heads, i.e., $\mathbf{m}_j = [\mathbf{m}_{1,j}; \mathbf{m}_{2,j}; ...; \mathbf{m}_{h,j}]$.

The third one is an additive attention network. Since the tweets written by users can be very noisy, and many words in tweets are uninformative to DN and ADR detection. For example, in the tweet "I like to work with adderall, but it will make me sleepless", the word "sleepless" is informative to the detection of ADR, but the word "work" is less informative. In order to attend the contexts which are informative to DN and ADR detection, we propose to use the additive attention mechanism in our approach to highlight the informative contexts for building the final representations of tweets. It takes the word representations as the input, and calculate a weight α_i for each word which reflects the importance of it. The attention weight α_i is evaluated by:

$$r_i = \tanh(\mathbf{u}_w^T \mathbf{m}_i + b_w), \tag{5}$$

$$\alpha_i = \frac{\exp(r_i)}{\sum_{k=1}^{M} \exp(r_k)}, \tag{6}$$

The final representations of a tweet are computed by:

$$\mathbf{r} = \sum_{k=1}^{M} \alpha_k \mathbf{m}_k. \tag{7}$$

3.3 Tweet Classification

The tweet classification module is used to classify whether a tweet contains a mention of DN or ADR. It uses a dense layer with ReLU activation to transform the hidden representations of tweets first, which is formulated as:

$$\mathbf{r}' = \mathrm{ReLU}(\mathbf{U}_r \mathbf{r} + \mathbf{b}_r), \tag{8}$$

where $\mathbf{U}_r$ and $\mathbf{b}_r$ are parameters. Then two separate dense layers with softmax activation function are used to predict the final label. The first one predicts whether a tweet mentions DN. We denote this task as *DN*. The second one predicts whether a tweet mentions ADR. We denote this task as *ADR*. The predicted labels $\hat{y}^D$ and $\hat{y}^A$ of a tweet in the *DN* and *ADR* tasks are respectively computed as:

$$\hat{\mathbf{y}}^D = softmax(\mathbf{U}_d^T \mathbf{r}' + \mathbf{b}_d), \tag{9}$$

$$\hat{\mathbf{y}}^A = softmax(\mathbf{U}_a^T \mathbf{r}' + \mathbf{b}_a). \tag{10}$$

Usually, tweets mentioning an ADR may also mentions DNs and vice versa. For example, the tweet "I took too much cocaine and I'm extremely sleepless" contains both DN and ADR mentions. Therefore, we can exploit the inherent relatedness between the DN and ADR tasks by training our model in the two tasks jointly. In our approach, the loss function $\mathcal{L}$ we use is formulated as follows:

$$\mathcal{L} = -\lambda \frac{1}{|\mathcal{S}_D|} \sum_{y^D \in \mathcal{S}_D} \sum_{k=1}^{2} y_k^D \log(\hat{y}_k^D) - (1-\lambda) \frac{1}{|\mathcal{S}_A|} \sum_{y^A \in \mathcal{S}_A} \sum_{k=1}^{2} y_k^A \log(\hat{y}_k^A), \tag{11}$$

where $\mathcal{S}_D$ and $\mathcal{S}_A$ denote the data sets in the *DN* and *ADR* tasks respectively, y_k^D and y_k^A are the gold labels in the two tasks, λ is a hyper-parameter to control the relative importance of the task *DN*.

4 EXPERIMENT

4.1 Dataset and Experimental Settings

We conduct experiments on two real-world datasets, i.e., the datasets provided by Task 1 and Task 3 in the shared tasks of the third SMM4H workshop[1] [33].

The first one is for the detection of tweets mentioning DNs (task *DN*). It contains 9,622 tweet IDs (4,975 positive and 4,647 negative samples) for training, and 5,382 tweets for test. The second one is for the detection of tweets mentioning ADRs (task *ADR*). It contains 25,598 tweet IDs (2,223 positive and 23,375 negative samples) for training, and 5,000 tweets for test. However, many tweets are not available now, we only crawled 9,065 (4,705 positive and 4,360 negative) and 16,694 (1,355 positive and 15,336 negative samples) tweets for training in *DN* and *ADR* tasks respectively using these IDs. Besides, since the gold labels of the test sets in the shared tasks are not available yet, we only use the aforementioned training sets in our experiments. We randomly sampled 80% of tweets for training, 10% for validation and 10% for test.

In our experiments, we use the 400-dim pre-trained word embeddings released by Godin et al. [8]. The Bi-LSTM network has 2×200 units. The CNN network has 200 filters with window size of 3. There are 24 heads in the multi-head self-attention network, and the output dimension of each head is 24. The loss weight λ is set to 0.5. RMSProp [4] is selected as the optimizer. The size of a mini-batch is set to 64. Since the negative samples are dominant in the ADR dataset, we use the over-sampling strategy [32] by repeating the positive samples for k times, and we set k to 9 in our approach. In our experiments, the hyper-parameters are selected via cross validation. The performance is evaluated by the precision, recall and F-score on the positive samples. We repeat each experiment independently for 10 times and report the average performance.

4.2 Performance Evaluation

In this section, we will evaluate the performance of our methods with several baseline methods on the two benchmark datasets. The methods to be compared are listed as follows:

- *SVM*: support vector machine using bag-of-word features, which is a widely used method for the detection of DN and ADR mentioning tweets [18, 25].
- *ME*: maximum entropy models using n-grams features. We use the codes released by Zhang et al. [34].
- *CNN*: convolutional neural networks [14, 17, 22].
- *LSTM*: bi-directional long short-term memory network [13].
- *CRNN*: the combination of CNN and LSTM [14].
- *RCNN*: the combination of LSTM and CNN [14].
- *Lee et al.* [21]: using CNN and additional unlabeled corpus to enhance the word embeddings.
- *CNN-Att*: using the combination of CNN and additive attention [14].
- *LSTM-Att*: using the combination of LSTM and additive attention [35].
- *Han et al.* [21]: using CNN and the concatenation of multiple additive attention networks.
- *MSA-basic*: our basic hierarchical tweet representation model without attention mechanism.
- *MSA*: our proposed model.

For fair comparisons, we use the same word embeddings and joint training methods in all baseline neural networks. We conducted experiments using different amount of training data (i.e., 10%, 25% and 100%) in *DN* and *ADR* tasks. The results are shown in Table 2. According to these results, we have several findings.

First, the methods based on neural networks consistently outperform *SVM* and *EM*. For example, our model *MSA* can achieve 90.4% and 52.4% in F-score on the DN and ADR datasets respectively, but *SVM* can only achieve 86.0% and 47.3% respectively. Since *SVM* and *EM* methods are based on bag-of-word features, they cannot effectively capture the contexts and the orders of words, which are both very important for the detection of DN and ADR mentioning tweets.

Second, the models using hierarchical tweet representations (*MSA-basic* and *MSA* outperform the flatten models (e.g., *CNN*, *LSTM*, *CRNN*, *RCNN* and Lee et al. [21]) in both *DN* and *ADR* tasks. Since there are massive misspellings and user-created abbreviations which are usually OOV, the representations of words can be enhanced by building from their characters directly, which may be beneficial to detect the DN and ADR mentioning tweets more accurately.

Third, models using attention mechanism (e.g., *CNN-Att*, *LSTM-Att* and *MSA*) outperform their variants without attention (e.g., *CNN*, *LSTM* and *MSA-basic*). Since tweets are often very noisy, there are many words that are uninformative to the detection of the DN and ADR mentioning tweets. Therefore, highlighting the informative contexts by incorporating the additive mechanism may help to build better tweet representations, which is beneficial for the detection of the DN and ADR mentioning tweets.

Fourth, although several existing methods such as *CNN-Att*, *LSTM-Att* and Han et al. [11] incorporate attention mechanism into DN and ADR detection models, our *MSA* approach can consistently

[1] https://healthlanguageprocessing.org/smm4h/social-media-mining-for-health-applications-smm4h-workshop-shared-task/

Table 2: The performance of different methods using different amount of training data on the DN and ADR datasets. *Denotes the statistical significance for p < 0.01 compared to the baselines.

Method	DN									ADR								
	10%			25%			100%			10%			25%			100%		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
SVM	76.1	74.8	75.4	83.4	82.1	82.8	87.0	85.1	86.0	34.1	37.0	35.5	41.8	43.5	42.6	46.7	48.0	47.3
ME	75.9	75.8	75.8	83.7	83.1	83.4	87.4	86.9	87.2	35.7	36.4	36.0	42.8	43.6	43.2	48.5	48.8	48.7
CNN	78.0	74.9	76.3	85.6	83.2	84.3	89.6	86.9	88.3	35.2	38.6	36.8	42.9	45.5	44.1	48.3	52.1	50.2
LSTM	77.0	75.3	76.0	84.9	83.6	84.2	89.0	87.3	88.1	36.1	38.2	37.0	43.5	45.2	44.3	49.6	50.5	50.0
CRNN	76.4	75.4	75.8	85.0	83.8	84.3	88.9	88.0	88.5	35.8	37.7	36.7	43.1	45.3	44.2	49.8	50.4	50.1
RCNN	76.9	76.0	76.4	85.5	83.7	84.5	89.0	88.3	88.6	36.5	38.3	37.3	43.7	45.6	44.5	47.5	53.4	50.2
Lee et al. [21]	77.8	75.4	76.5	85.2	83.5	84.3	89.1	88.2	88.6	36.5	38.1	37.2	44.0	44.9	44.4	49.1	51.6	50.3
CNN-Att	78.4	75.1	76.6	86.6	83.1	84.8	90.3	87.0	88.7	35.1	39.4	37.2	42.7	46.5	44.7	48.3	53.0	50.5
LSTM-Att	77.2	75.8	76.4	85.9	83.6	84.5	89.4	87.7	88.5	36.0	38.9	37.4	43.2	46.6	44.6	48.5	52.7	50.5
Han et al. [11]	77.5	75.4	76.4	85.2	84.3	84.7	89.4	88.3	88.8	36.3	38.6	37.4	43.9	45.3	44.6	49.8	51.0	50.4
MSA-basic	78.0	75.6	76.7	86.1	84.4	85.3	90.2	87.9	88.9	35.7	39.6	37.5	42.8	46.4	44.7	48.8	52.7	50.6
MSA*	**80.4**	**77.3**	**78.8**	**87.5**	**85.7**	**86.6**	**91.2**	**89.5**	**90.4**	**37.7**	**42.2**	**39.9**	**45.0**	**48.8**	**46.8**	**51.2**	**53.8**	**52.4**

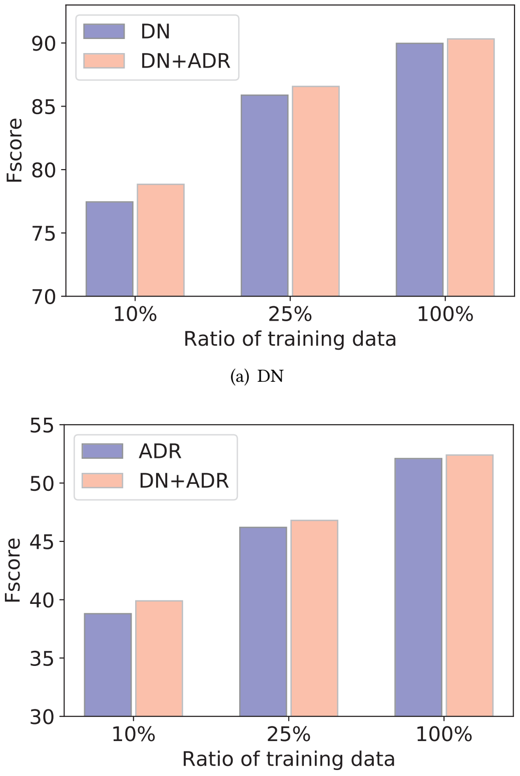
outperform them. In these baseline methods, the attention weight of a word is computed only based on its hidden representation, and the relationships between different words in a text cannot be modeled. Different from these methods, our approach can take advantage of both self-attention and additive attention. Usually global contextual information is very important for detecting tweets mentioning DNs and ADRs. Our approach can effectively model the interactions between a word with all other words in a tweet using the multi-head self-attention network, which may be very useful to the detection of the DN and ADR mentioning tweets.

4.3 Effectiveness of Joint training

In this section, we will validate the effectiveness of joint training, i.e., training our model simultaneously in both DN and ADR tasks. We conduct experiments using our proposed model *MSA* under different amount of training data. The experimental results on the two datasets are shown in Figure 2(a) and 2(b). According to the results, we have several findings. Compared with the models trained in the task *DN* or task *ADR* only, using the combination of both tasks can consistently improve the performance. Since the tweets mentioning ADRs may mention DNs and vice versa, the two tasks have inherent relatedness and can share rich mutual information. Therefore, capturing the useful mutual information in both tasks may be beneficial for building better representations of words and tweets, which can improve the performance in both tasks. In addition, we find that the improvement brought by joint training is more significant when training data is more scarce. Since manual annotation in the DN and ADR detection tasks is very expensive and time-consuming, training both tasks jointly can also help to reduce the efforts on manual annotation.

4.4 Effectiveness of Attention Mechanism

In this section, we will validate the effectiveness of attention mechanism in our approach, i.e., the multi-head self-attention and additive attention. We compare the performance of our approach *MSA* and its variants (i.e., without attention, with additive attention and with multi-head self-attention). The experimental results on the two datasets are illustrated in Figure 3(a) and 3(b).

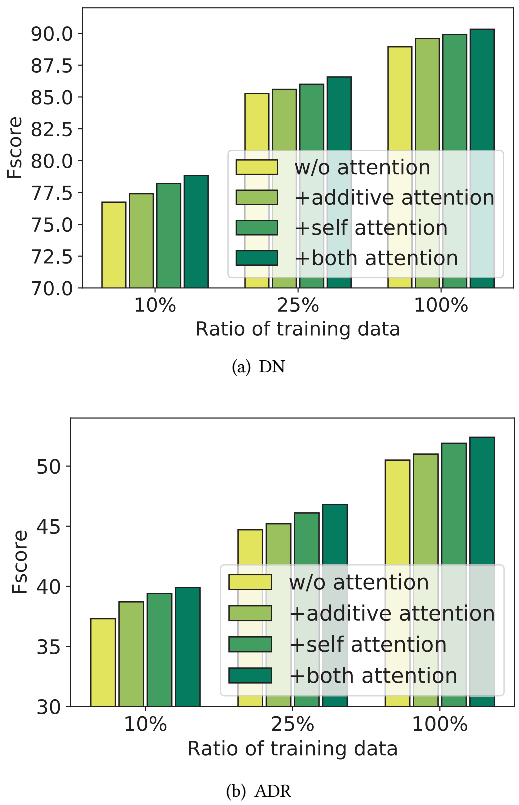


(a) DN

(b) ADR

Figure 2: The performance of our approach in Fscore with joint training or not.

According to the results, the multi-head self-attention mechanism can effectively improve the performance. This is probably because the self-attention mechanism can model the interactions of a word between all words in texts, and global contextual information is useful for more accurate DN and ADR mentioning tweets detection. In addition, the additive attention can also improve the



(a) DN

(b) ADR

Figure 3: Influence of attention mechanism on the performance of our approach in F-score.

performance consistently. Since some words can be important indications of DN and ADR mentions, focusing on these informative words within tweets can help to build more informative representations of tweets, which may be beneficial for the detection of DN and ADR mentioning tweets. Moreover, combining both self-attention and additive attention can achieve a better performance. It indicates that taking advantage of both types of attention networks can detect DN and ADR mentions more accurately by building better representations of tweets and their words.

4.5 Influence of Hyper-parameters

In this section, we will explore the influence of two important hyper-parameters on our approach, i.e., the over-sampling rate k in the *ADR* task and the loss weight λ in Eq. (11).

The over-sampling rate k aims to control the ratio of positive samples in the training set. We conducted experiments using different over-sampling rate k. The experimental results are shown in Figure 4. According to the results, we have several findings. When the over-sampling rate k is too small, negative samples are dominant in the training data, which will lead to a poor recall. Thus, the performance in F-score will be sub-optimal. When the over-sampling rate k is too large, the positive samples will be over-emphasized and the precision will decline. Therefore, the performance in F-score will also be sub-optimal. Besides, we find it's interesting that balancing the number of positive and negative samples ($k = 11.3$) directly is not optimal, and using a lower k (e.g., between 7 and 10) can achieve a better performance. It may be because the diversity of training data will become low if the positive samples are repeated too many times, and the model may intend to predict a tweet as a positive sample, which may lead to the sub-optimal performance.

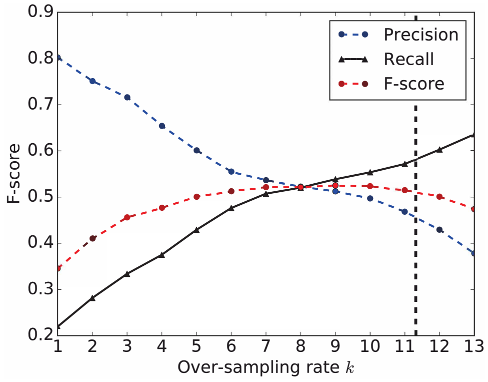


Figure 4: Influence of the over-sampling rate k on our approach in the ADR task. The black dashed line denotes the original ratio of negative samples to positive samples.

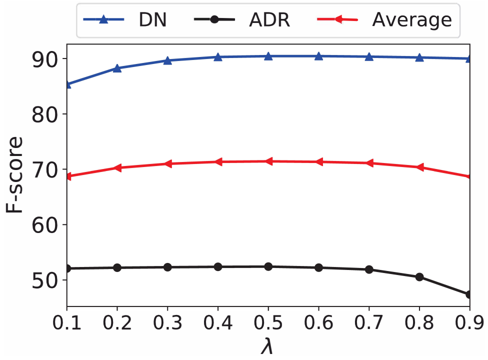
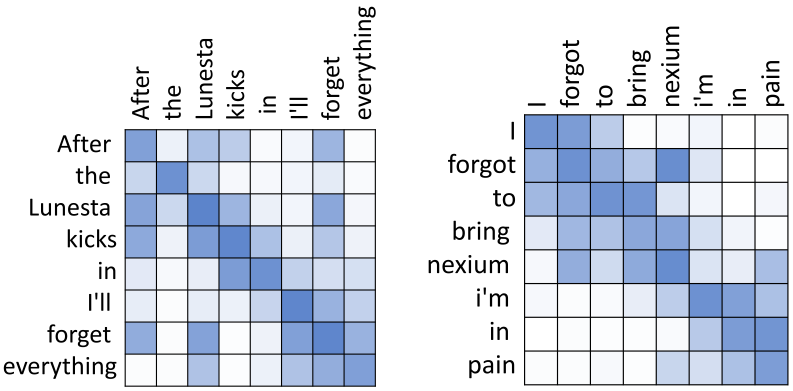


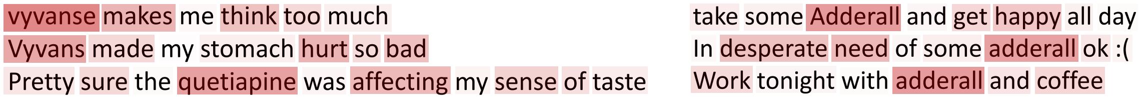
Figure 5: Influence of the loss weight λ on the performance of our approach in both tasks.

The loss weight λ aims to control the relative importance of the *DN* and *ADR* tasks. The experimental results on λ are shown in Figure 5. In order to evaluate the overall performance of our approach, the average performance of the two tasks in F-score is also reported in Figure 5. According to Figure 5, we find the average performance of our approach improves when λ increases. This is because when λ is too small, the model will pay little attention to the *DN* task and its useful information cannot be fully exploited, which will lead to sub-optimal performance. However, when λ is too large, the average performance will also decline. This is because the *DN* task is over-emphasized and the *ADR* task will gain little attention, which will also lead to sub-optimal performance. Therefore, a moderate setting of λ (e.g., $\lambda = 0.5$) may be more appropriate.



(a) ADR mentioning tweet (b) non-ADR mentioning tweet

Figure 6: Visualization of the self-attention mechanism. Darker colors represent higher attention weights.



vyvanse makes me think too much
Vyvans made my stomach hurt so bad
Pretty sure the quetiapine was affecting my sense of taste

take some Adderall and get happy all day
In desperate need of some adderall ok :(
Work tonight with adderall and coffee

(a) ADR mentioning tweet (b) non-ADR mentioning tweet

Figure 7: Visualization of the additive attention. Darker colors represent higher attention weights.

4.6 Case Study

In this section, we will conduct several case studies to visually explore the influence of self-attention and additive attention mechanism on our approach. First, we will explore the influence of self-attention mechanism. The visualization results of a head in the self-attention network are shown in Figure 6. Darker colors denote higher attention weights, i.e., the interaction between the two corresponding words is more important. From the results, we find that our model can mine the important interactions between words. For example, in Figure 6(a), our model can find the word "Lunesta" interacts with the words "after", "kicks" and "forget", which are all important to infer whether this tweet mentions an ADR. Since global contextual information is usually important to the detection of DN and ADR mentioning tweets, modeling the interactions between all words in tweets can help to build better contextual representations of words and improve the performance of our model in the DN and ADR detection tasks.

Then, we will explore the influence of additive attention. For instance, the visualization results of additive attention in the *ADR* task are illustrated in Figure 7. The results show that incorporating additive attention can help to select informative contexts that contribute to the detection of ADR mentioning tweets. For example, the drug name "vyvanse" gains a high attention weight since it is very informative, while "me" receives little attention since it may be uninformative. In Figure 7(a), the word "vyvanse" is an important clue of ADR mentions. However, it is misspelled as "Vyvans" in the second example. Luckily, since the representations of words can be built directly from characters, the additive attention network can still recognize it is important for ADR detection and assign a relative high attention weight to it. It shows that our approach is robust to the massive noisy texts in tweets.

5 CONCLUSION

Detecting the tweets mentioning drug names and adverse drug reactions is an interesting and important task in the data mining field. In this paper, we propose a neural approach to jointly detect tweets mentioning drug names or adverse drug reactions. In order to alleviate the influence of massive misspellings and user-created abbreviations in tweets, we propose to use a hierarchical tweet representation model to first learn representations of words from characters and then build representations of tweets from words. In addition, we propose to use multi-head self-attention mechanism to fully capture the contexts in tweets by modeling the interactions between words. Besides, we incorporate the additive attention mechanism to select the informative words in tweets to build more informative representations of tweets. Experimental results on the benchmark datasets validate the effectiveness of our approach in detecting tweets mentioning DNs and ADRs.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of this paper. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800402 and in part by the National Natural Science Foundation of China under Grant U1705261, Grant U1536207, Grant U1536201 and U1636113.

REFERENCES

[1] Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, Kayo Waki, and Kazuhiko Ohe. 2010. Extraction of adverse drug effects from clinical records.. In *MedInfo*. 739–743.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Anne Cocos, Alexander G Fiks, and Aaron J Masino. 2017. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts. *Journal of the American Medical Informatics Association* 24, 4 (2017), 813–821.

[4] Yann Dauphin, Harm de Vries, and Yoshua Bengio. 2015. Equilibrated adaptive learning rates for non-convex optimization. In *NIPS*. 1504–1512.

[5] Andrea Esuli and Fabrizio Sebastiani. 2007. SentiWordNet: a high-coverage lexical resource for opinion mining. *Evaluation* 17 (2007), 1–26.

[6] Carol Friedman. 2009. Discovering novel adverse drug events using natural language processing and mining of the electronic health record. In *Conference on Artificial Intelligence in Medicine in Europe*. Springer, 1–5.

[7] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 315–323.

[8] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. In *WNUT*. 146–153.

[9] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610.

[10] Harsha Gurulingappa, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2011. Identification of adverse drug event assertive sentences in medical case reports. In *KD-HCM, ECML PKDD*. 16–27.

[11] Sifei Han, Tung Tran, Anthony Rios, and Ramakanth Kavuluru. 2017. Team UKNLP: Detecting ADRs, Classifying Medication Intake Messages, and Normalizing ADR Mentions on Twitter.. In *SMM4H@ AMIA*. 49–53.

[12] Rave Harpaz, William DuMouchel, Nigam H Shah, David Madigan, Patrick Ryan, and Carol Friedman. 2012. Novel data-mining methodologies for adverse drug event discovery and analysis. *Clinical Pharmacology & Therapeutics* 91, 6 (2012), 1010–1021.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[14] Trung Huynh, Yulan He, Alistair Willis, and Stefan Rueger. 2016. Adverse Drug Reaction Classification With Deep Neural Networks. In *COLING Technical Papers*. 877–887.

[15] Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in neural information processing systems*. 919–927.

[16] JITENDRA Jonnagaddala, Toni Rose Jue, and Hong-Jie Dai. 2016. Binary classification of Twitter posts for adverse drug reactions. In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing, Big Island, HI, USA*. 4–8.

[17] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[18] Svetlana Kiritchenko, Saif M Mohammad, Jason Morin, and Berry de Bruijn. 2018. NRC-Canada at SMM4H Shared Task: Classifying Tweets Mentioning Adverse Drug Reactions and Medication Intake. *arXiv preprint arXiv:1805.04558* (2018).

[19] Michael Kuhn, Monica Campillos, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. 2010. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology* 6, 1 (2010), 343.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.

[21] Kathy Lee, Ashequl Qadir, Sadid A Hasan, Vivek Datla, Aaditya Prakash, Joey Liu, and Oladimeji Farri. 2017. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 705–714.

[22] Arjun Magge, Matthew Scotch, and Graciela Gonzalez. 2017. CSaRUS-CNN at AMIA-2017 Tasks 1, 2: Under sampled CNN for text classification. In *CEUR Workshop Proceedings*, Vol. 1996. 76–78.

[23] Vassilis Plachouras, Jochen L Leidner, and Andrew G Garrow. 2016. Quantifying self-reported adverse drug events on Twitter: signal and topic analysis. In *Proceedings of the 7th 2016 International Conference on Social Media & Society*. ACM, 6.

[24] Majid Rastegar-Mojarad, Ravikumar Komandur Elayavilli, Yue Yu, and Hongfang Liu. 2016. Detecting signals in noisy data-can ensemble classifiers help identify adverse drug reaction in tweets. In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing*.

[25] Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics* 53 (2015), 196–207.

[26] Abeed Sarker and Graciela Gonzalez-Hernandez. 2017. Overview of the second social media mining for health (smm4h) shared tasks at amia 2017. *Training* 1, 10,822 (2017), 1239.

[27] Abeed Sarker, Azadeh Nikfarjam, and Graciela Gonzalez. 2016. Social media mining shared task workshop. In *Biocomputing 2016: Proceedings of the Pacific Symposium*. World Scientific, 581–592.

[28] Hashim Sharif, Fareed Zaffar, Ahmed Abbasi, and David Zimbra. 2014. Detecting adverse drug reactions using a sentiment classification framework. (2014).

[29] Richard Sloane, Orod Osanlou, David Lewis, Danushka Bollegala, Simon Maskell, and Munir Pirmohamed. 2015. Social media and pharmacovigilance: a review of the opportunities and challenges. *British journal of clinical pharmacology* 80, 4 (2015), 910–920.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[31] Xiaoyan Wang, George Hripcsak, Marianthi Markatou, and Carol Friedman. 2009. Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study. *Journal of the American Medical Informatics Association* 16, 3 (2009), 328–337.

[32] Gary M Weiss, Kate McCarthy, and Bibi Zabar. 2007. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *DMIN* 7 (2007), 35–41.

[33] Davy Weissenbacher, Abeed Sarker, Michael Paul, and Graciela Gonzalez-Hernandez. 2018. Overview of the Third Social Media Mining for Health (SMM4H) Shared Tasks at EMNLP 2018. In *EMNLP*.

[34] Zhifei Zhang, JY Nie, and Xuyao Zhang. 2016. An ensemble method for binary classification of adverse drug reactions from social media. In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing*.

[35] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based LSTM network for cross-lingual sentiment classification. In *EMNLP*. 247–256.